Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

Collaborative Intrusion Detection leveraging Blockchain and Pluggable Authentication Modules

Vikram Kanth, Ashley McAbee, Murali Tummala, and John C. McEachen
Naval Postgraduate School
{vkkanth, asmcabee1, mtummala, mceachen}@nps.edu

Abstract

As the threat of cyber attack grows ever larger, new approaches to security are required. While there are several different types of intrusion detection systems (IDS), collaborative IDS (CIDS) offers particular promise in identifying distributed, coordinated attacks that might otherwise elude detection. Even for CIDS, there are unresolved issues associated with trusting participants and aggregating data. Blockchain technology appears capable of addressing those issues if practical implementation strategies can be developed. To that end, we implement an Ethereum blockchain-based CIDS leveraging pluggable authentication modules. Our system is specifically crafted to detect doorknob rattling attacks by immutably recording login activity in a blockchain-protected ledger.

1. Introduction

There is no doubt that cyber attacks are an ever-growing threat. Statistics collected by the U.S. Government Accountability Office (GAO) indicate that the number of reported attacks against federal agencies has steadily increased by an average of 8,000 attacks per year for more than a decade [1], the Department of Defense estimates that 36 million email attacks take place against defense infrastructures every day [2], and an estimated 10% of U.S. residents over the age of 16 were victims of cyber-perpetrated identity theft in 2016 alone [3]. Surveying this threat landscape, there is a clear need for more effective defensive tools.

One of the most commonly used defenses is intrusion detection [4]. The objective of such systems is to recognize anomalous behavior either within the network as a whole or within individual hosts. Although there are variations, the work flow illustrated in Figure 1 is representative of the basic tasks IDS accomplish en route to flagging anomalies.

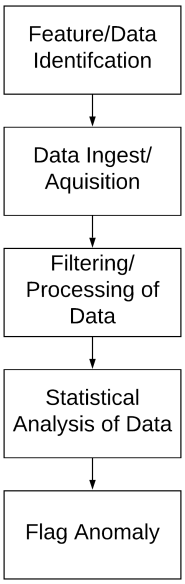

Figure 1. General Flow Chart for Intrusion Detection

The flow begins with feature identification and ingest into the system before processing and analysis facilitate anomaly detection. Networked-based IDS monitor network-wide behavior, for example traffic volume or diversity, in order to detect the changes in patterns that indicate an attack. In contrast, host-based IDS observe local events like log-in attempts or unusual privilege escalation in an effort to flag undesirable activity. While these approaches are effective in monitoring a single system or network, they are unable to detect highly distributed attacks due to the lack of communication amongst stand-alone IDS across a major network or system [5]. To overcome this shortfall, hybrid approaches that blend host-based monitoring, network-based statistics, and aggregate reporting of both across sets of nodes have emerged. This is the realm of Collaborative IDS (CIDS).

CIDS employ sensors to pool data from across the protected network for follow on anomaly detection

URI: https://hdl.handle.net/10125/64564
978-0-9981331-3-3

analysis. Some touted advantages of CIDS include increased detection accuracy and significantly reduced work for security administrators [6].

CIDS are classified as centralized, hierarchical, or distributed. Specific implementation details of these various types of CIDS are covered in greater detail by Vasilomanolakis *et al.* [5], but the disadvantages of centralized and hierarchical systems hinge on their dependency on just a few nodes [7]. Conversely, distributed CIDS overcome this dependency by spreading the data collection and analysis tasks across the network. However, this architecture also has implementation issues related to creating consensus and trust between peered nodes in the distributed system. To facilitate wider adoption of distributed CIDS in light of its many advantages [5], we explore leveraging blockchain technology to address challenges of trust and consensus.

Many industries are exploring how blockchain technology might improve their processes; cyber security is no different. We offer a basic overview of blockchain technology to highlight the tenets important for our proposed application, but further detail regarding blockchain is beyond the scope of this paper and can be found in references [8] and [9]. The critical component of blockchain technology for CIDS applications is its mechanism of validating and storing data with no need for a central, trusted authority.

When data, sometimes called transactions because of blockchain technology's cryptocurrency roots, are ready for storage, they are sent to all participating nodes for consensus-based validation. Groups of validated data form blocks, which are completed via the addition of a cryptographic hash linking them to the full chain of previously archived data. The process of developing that hash is often called *mining*. The crucial benefit of these cryptographic hashes is that once a block is added to the chain and approved via distributed consensus, it cannot be removed or edited without overcoming significant cryptographic barriers simultaneously at a consensus-inducing portion of the nodes.

For a collaborative intrusion detection system that leveraged blockchain to facilitate sharing of system logs, this means an attacker would have to simultaneously overcome computationally expensive cryptographic barriers on a large number of separately secured machines to erase the record of their actions. There are three types of blockchain ledgers currently in use: public, consortium, and private [10]. Public blockchain systems allow anyone with internet access and a desire to participate to do so. Consortium blockchain systems are maintained by an established body that grants access to others. Private blockchain systems are maintained by one entity that provides permissions to others. More detailed information can be found in [10]. Depending on the CIDS use case, different ledgers might apply.

Our focus in this research effort is to address trust and consensus in the data ingestion phase of CIDS. We investigate the feasibility of overlaying a blockchain network on an existing network to facilitate trustworthy data collection in support of intrusion detection. Building on the conceptual blockchain-based CIDS introduced by Alexopolous et al [7], we propose a specific and practical implementation.

The primary contribution of this paper is the characterization of a CIDS leveraging a private Ethereum-hosted blockchain and pluggable authentication modules (PAM) to create a distributed collaborative intrusion detection system capable of flagging some types of attacks. The system is novel is its use of PAM to log authentication events as they are occurring. The combination of PAM with blockchain technology's distributed immutable ledger offers a unique approach to preventing attackers from modifying system logs in an attempt to obfuscate malicious activity.

The remainder of the paper is organized as follows: Section 2 details the problems in distributed collaborative intrusion detection and how blockchain technology addresses those concerns, Section 3 provides our solution approach, and Section 4 details the specifics of our implementation.

2. Improving data ingest in CIDS via blockchain

Each of the tasks in Figure 1 is a rich area for investigation with a variety of complex approaches available. Our research focus is improving the data ingest and acquisition process. Imagine a scenario in which a network is comprised of several nodes where each monitors and reports various events. The confidence in the anomaly detection capabilities of the system as a whole begins with confidence in the accuracy and completeness of the record of events of interest.

The problem of maintaining accuracy and completeness is even more difficult when a coordinated attack takes place [5]. When coordinated attacks are choreographed to maintain a low volume or rate at any one host, respectful of common intrusion detection activity thresholds, the network must be able to aggregate events to be able to respond. Consider the example of the doorknob rattling attack on a Supervisory Control and Data Acquisition system wherein an adversary attempts common login/password

combinations on several machines in the network, maintaining a low number of login attempts per device [11]. For the attack to succeed, the adversary need only gain remote access to one of these devices [11]. This type of attack is difficult to detect without CIDS. As evidence, consider the nominal log traces presented in Figure 2.

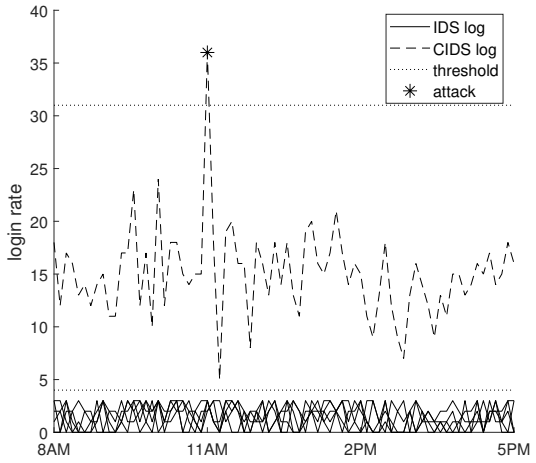

Figure 2. Nominal doorknob rattling attack detection via CIDS. The CIDS compiles reports from individual IDSs to track network activity as the sum of reported activity.

In this scenario, ten stand alone IDS are tracking login rate, with a CIDS aggregating login data across the system as a whole. If either the local or aggregate login rate is larger than a specified threshold, an alert would be triggered. For the ten stand alone IDS expecting to record between zero and three logins per reporting period, a detection threshold of unusual activity might be set at four. Still, on average, half would have ignored the 11:00am doorknob rattling attack as it involves just two extra login attempts per system. This activity level disappears in the normal network activity on any one node. In contrast, the attack clearly stands out in the CIDS log trace, easing the complexity required in the follow on data processing and analysis phases of anomaly detection.

Blockchain technology brings an added benefit in log preservation. If an adversary is successful in accessing a more traditional, centralized CIDS, they could alter the access log to remove evidence of the attack, as hackers often tamper with evidence of their presence on a system to thwart future forensic analysis [12], but editing the cryptographically protected, distributed log stored via blockchain on all ten nodes is significantly more difficult.

3. Proposed blockchain-based approach

The doorknob rattling scenario from Section 2 showed the need for a collaborative system to detect certain attacks. Vasilomanolakis *et al.* [5] and Alexopolous *et al.* [7] lay out the principal requirements of such a system, which are compiled in Table 1.

Table 1. CIDS Requirements, compiled from [5, 7]

Accountability	Actions taken by a participating node must be tied to that node.
Integrity	Data cannot be modified once entered into the system.
Resilience	The system should not depend on small numbers of node and must avoid single points of failure.
Consensus	Nodes in the system must reach consensus about the quality and trustworthiness of data.
Scalability	The system must be able to scale to larger numbers of agents.
Overhead	Overhead must be minimized to allow flexibility of the system.
Privacy	Depending on the type of system, some level of privacy must be considered for participants.

There are clear trade-offs between the attributes specified in the table. For example, there are often conflicts between privacy and accountability [7]. Furthermore, depending on where such a system would be utilized, the requirements shift. For example, database entries with private customer account details require different privacy considerations than logs of privilege escalation on a controlled system. Framing the design of CIDS in the context of these requirements is useful toward capturing the key characteristics of the system while minimizing overhead and cost. For example, overlaying data encryption using a public key infrastructure necessarily increases overhead and thus should be implemented only when needed [7].

The blockchain-based system loosely described in Section 2 fulfills the requirements in Table 1, with support as follows: Because each transaction contains a sender and a receiver that cannot be modified once added to the chain, the requirement of accountability is met. Further, once approved transactions are added to the chain they cannot be modified without overcoming significant cryptographic barriers [9], fulfilling the requirement of integrity. Resilience and consensus are both met via blockchain's embedded distributed consensus mechanisms [8], which

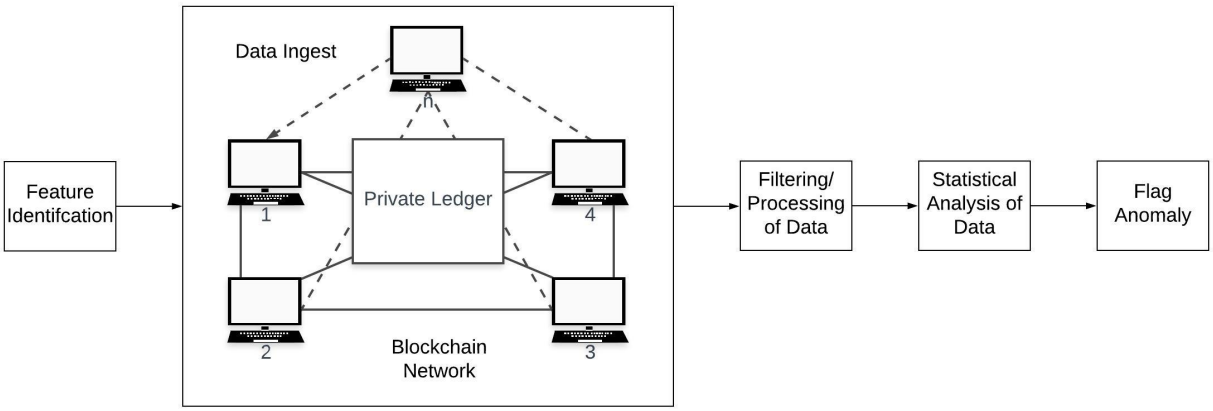

Figure 3. Blockchain Solution for CIDS

can be implemented to ensure single points of failure are avoided and graceful degradation is possible. As evidenced by wide use of cryptocurrencies like Bitcoin and Ethereum, blockchain is scalable with appropriate implementation considerations. The cryptocurrency use case also demonstrates that privacy can be achieved in blockchain-based systems via the assignment of non-attributable identities.

As Alexopoulous *et al.* [7] note, overhead can be a problem for blockchain systems, but it is dependent on consensus mechanism and other design decisions and can be mitigated via techniques like alert hashes and bloom filters. To satisfy these requirements, Alexopoulous *et al.* [7] propose a CIDS framework utilizing a secure distributed ledger implemented by blockchain technology to exchange alerts between collaborating nodes. These alerts become the transactions of the blockchain system. Depending on the types of attacks or behaviors that an organization is worried about, different metrics must be utilized. A more detailed discussion on different metrics can be found in [13]. As a few key examples, finding an adversary running unwanted programs might require logging CPU utilization statistics. Source and destination ports for data access requests could be another potential metric of interest. To thwart the doorknob rattling attack from Section 2, the transactions should include all login attempts.

We develop a CIDS to thwart the doorknob rattling attack by leveraging blockchain technology to facilitate a distributed ledger of login and authentication attempts aggregated across a network. The basic architecture of the system is displayed in Figure 3.

We make a few assumptions in this general approach. First, for most organizations, a private ledger is the most logical mechanism for this log, with the system administrator validating participant identity upon initialization of the system, at a minimum, which leads to our second assumption that we have a priori knowledge of participants on the network. Although the ledger in Figure 3 is pictured in the center of the nodes, this is only to emphasize that consensus is maintained on the ledger's contents; it is physically stored at every node to maintain the distributed attributes important to the system's benefits. Data from the private ledger can be used for processing and statistical analysis much like in any other IDS, with the specific anomaly detection mechanism outside the scope of this paper.

In the solution proposed in [7], nodes are categorized as either monitoring units or analysis units. Any node in the blockchain network can be either although the technical requirements for each will be different. Some systems may also require two layers of communication: an alert layer and a consensus layer to allow flexibility in scenarios where permissioned viewing lists are necessary to achieve required privacy levels [7].

The range of scenarios that the system must handle will heavily influence design decisions, but to focus our initial investigation, we considered the case illustrated in Figure 3, where all nodes participate fully as both monitoring and analysis units.

Our CIDS leverages commercially available and open-source products. The Ethereum platform provides our blockchain-based distributed ledger via the *testnet* functionality. More specific information on Ethereum can be found in [14]. We leveraged the standard *go-ethereum* client and the private test network functionality for all experiments. Our test bed consists of two Linux Ubuntu 18.04.1 systems with the Ethereum client running on both, as depicted in Figure 4. Although more work is needed to specifically quantify the performance requirements for blockchain-based CIDS,

our test bed demonstrates that at least for small scale implementations, generic hardware is adequate to run the CIDS with sufficient capacity preserved for general computing.

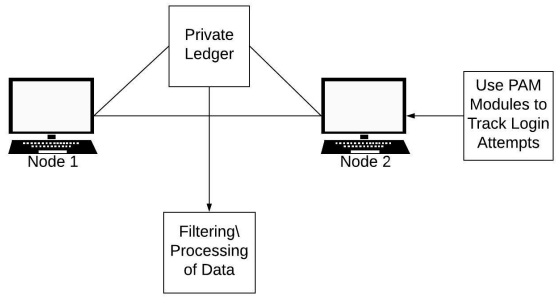

Figure 4. CIDS set up used in experiments

A sample transaction from our test network is shown in Figure 5. The *input* field carries the critical login data that we wish to permanently preserve as a hex string. The other fields deal with the nuts and bolts of cyrptocurrency transactions and are beyond the scope of this paper, refer to [14] for further explanation.

Figure 5. Example Ethereum Transaction

For the Linux operating system, including the Ubuntu 18.04.1 distribution used in our test network, PAM offer a suite of shared libraries to facilitate local control over how applications authenticate users [15]. When a user attempts to login, our system goes through the typical Linux authentication process but makes an additional call to *pam_exec*, which permits the system to run an external command. Note that the call to the *pam_exec* module must be put into the correct PAM service file (in Ubuntu the service file location is */etc/pam.d*). We used the *common-auth* service to call a Bourne Again SHell (Bash) script that interacted with our Ethereum client. The *pam_exec* module sets several environmental variables that record several important pieces of information for logging purposes. A subset of these variables are shown in Table 2. By altering the *common-auth* file, we ensured that every time an authentication request was made, it would be logged. This not only includes initial login attempts, but any other service that requires user authentication, such as the use of *sudo* to escalate privilege. The *pam_exec* module allowed us to run an external Bash script that had access to all of these environment variables.

That Bash script calls a Python script, feeding it those variables. We interact with the Ethereum client though the Web3 library in Python. Using the information passed by *pam_exec* and our Bash script, a node in our CIDS submits a transaction wherein the *data* field contains a hex string encoding the user, service, time, and type of authentication. Once a mining operation takes place to produce a new block, the transaction is codified into the ledger and can be viewed by any node in the CIDS.

Table 2. PAM Items Exported as Environment Variables by pam_exec, compiled from [15]

PAM_RHOST	Remote user attempting to authenticate.
PAM_RUSER	Remote host that is being authenticated to.
PAM_SERVICE	Service module that made request.
PAM_USER	User that made request
PAM_TYPE	Type of module (account, auth, password, open-session, close-session)

4. Validation and Results

With the CIDS in place, we simulated a doorknob rattling attack against one of our machines. In this test, the victims were different user accounts on a single machine. Output from our CIDS ledger is shown in Figure 6. The attacking machine (172.20.157.112) attempted to login to each victim user account three times via secure shell (SSH). With our modified *common-auth* file in place, the CIDS recorded each login attempt as a separate transaction in the blockchain as depicted in Figure 6. The logged information is summarized in Table 3.

This attack took place over 46 seconds, with 15 total login attempts across five different accounts. As this ledger is visible from any CIDS node, extraction of these login attempts for follow on analysis toward anomaly detection is possible from any of the distributed participants. Although beyond the scope of the current experiment, our system might have then responded by blocking any subsequent traffic from the attack-related

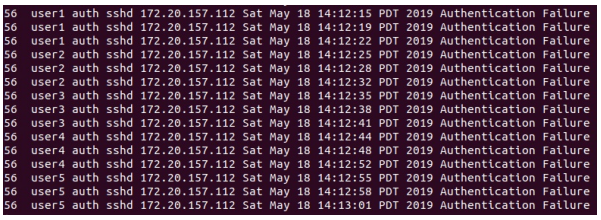

Figure 6. Doorknob rattling attack in ledger

Table 3. Summary of doorknob rattling attack events, single attacker

User	IP Address of Request	Number of Login Attempts	Duration of Attack (seconds)
user1	172.20.157.112	3	7
user2	172.20.157.112	3	10
user3	172.20.157.112	3	9
user4	172.20.157.112	3	11
user5	172.20.157.112	3	9

internet protocol (IP) address.

In a second experiment, multiple attackers acted against our victim machine. Output from the ledger in this case is shown in Figure 7. This scenario contained two attackers that acted in differing ways, as revealed by the detail the ledger contains. The first attacker (172.20.148.85) made all of its SSH requests simultaneously whereas the second attacker (172.20.157.112) made its SSH requests in sequence. This simple difference can reveal some information about the attacking machine. The first attacker was a Windows machine using the puTTY program for its SSH requests. The second attacker was another Ubuntu machine using OpenSSH to conduct its login attempts. The differing nature of these programs can be seen from the log entries in Figure 7. A summary of the attack is shown in Table 4. There were a total of 24 login attempts via two attackers across a time interval of 43 seconds.

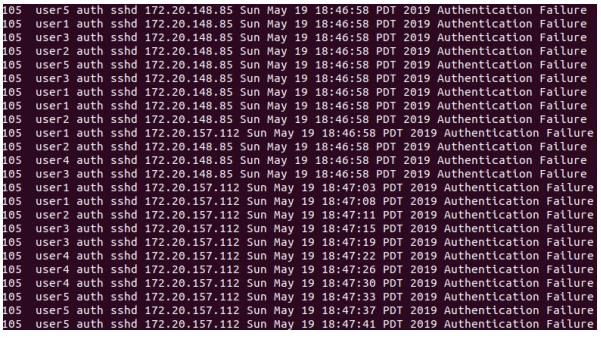

Figure 7. Multi-attacker doorknob rattling attack in ledger

Table 4. Summary of doorknob rattling attack events, multi-attacker

User	IP Address of Request	Number of Login Attempts
user1	172.20.157.112	3
user2	172.20.157.112	1
user3	172.20.157.112	2
user4	172.20.157.112	3
user5	172.20.157.112	3
user1	172.20.148.85	3
user2	172.20.148.85	3
user3	172.20.148.85	3
user4	172.20.148.85	1
user5	172.20.148.85	2

In both experiments, indications of the attack were submitted for inclusion in the CIDS distributed ledger within approximately 1.10 seconds of attack initiation, well faster than log entries could be manually modified by an attacker. This rapid indication of the event and protection of related data could facilitate action in time to protect other network nodes in near real time and also ensures a trustworthy forensic record of the attack events is preserved for follow on analysis. Although a specialized system might improve functionality and efficiency, we have concluded that Ethereum's unaltered *testnet* blockchain client integrates smoothly with existing Linux system architecture to accomplish data ingest toward intrusion detection with minimal adjustment or overhead and sufficient fidelity.

5. Conclusion

We have implemented a CIDS capable of immutably recording login activity via a private blockchain-based ledger. Our initial experiments show that blockchain-based CIDS is a viable method for detection of doorknob rattling attacks, preserving activity records more quickly than an intruder could act to modify them. These positive indications validate continued research.

The system engineered in this paper is small in scale. Creating a larger testbed would be the first step in further exploration of our approach. Additional research must also be conducted to determine how the system would work at scale with several types of devices. One issue that would need to be addressed as the system increases in scale is the network overhead to support the blockchain client. Developing an understanding of the overhead of a blockchain client must be a priority for this research to move forward.

Further, the system developed in this paper utilized the Ethereum client. This client was designed to support cryptocurrency functionality; a CIDS-specific blockchain apparatus might differ significantly in terms of consensus algorithm, transaction formatting, and other details. Work is needed to explore what that customization should entail. If designing a CIDS-specific blockchain system proves too high of a hurdle, additional research into which general purpose blockchain client best meets CIDS requirements should be conducted.

Finally, more work is also necessary to explore how this blockchain-based data ingest module will interact with follow on filtering and processing of data and the other remaining steps in the intrusion detection process outlined in Figure 1. There is potential for the blockchain system itself to provide enhanced detection of attacks. For example, an increased activity level in the blockchain could indicate some attack without necessitating inspection of individual transactions.

The sharing of information between cyber defenders has become crucial toward preventing distributed attacks against the system as a whole. CIDS are one way in which to address this need, and blockchain technology appears to be well-suited for the task of ingesting data in a distributed and secure fashion. While it is true that blockchain is not a solution to every type of problem, it shows promise in this area.

References

[1] U.S. Government Accountability Office, "Information security: Agencies need to improve controls over selected high-impact systems," Washington, DC, USA, GAO Report No. GAO-16-501, 2016.

[2] F. Konkel, "Pentagon thwarts 36 million email breach attempts daily," Jan 2018. [Online]. Available: https://www.nextgov.com/cybersecurity/2018/01/pentagon-thwarts-36-million-email-breach-attempts-daily/145149/

[3] U.S. Government Accountability Office, "Data breaches: Range of consumer risks highlights limitations of identity theft services," Washington, DC, USA, GAO Report No. GAO-19-230, 2019.

[4] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, pp. 805–822, 1999.

[5] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 55:1–55:33, May 2015. [Online]. Available: http://doi.acm.org/10.1145/2716260

[6] J. R. Goodall, W. G. Lutters, and A. Komlodi, "I know my network: Collaboration and expertise in intrusion detection," in *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '04. New York, NY, USA: ACM, 2004, pp. 342–345. [Online]. Available: http://doi.acm.org/10.1145/1031607.1031663

[7] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," in *Critical Information Infrastructures Security*, G. D'Agostino and A. Scala, Eds. Cham: Springer International Publishing, 2018, pp. 107–118.

[8] A. Baliga, "Understanding blockchain consensus models," Persistent Systems, Santa Clara, California, USA, 2017. [Online]. Available: https://www.persistent.com/whitepaper-understanding-blockchain-consensus-models/

[9] S. Singh and N. Singh, "Blockchain: Future of financial and cyber security," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Dec 2016, pp. 463–467.

[10] H. Okada, S. Yamasaki, and V. Bracamonte, "Proposed classification of blockchains based on authority and incentive dimensions," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, Feb 2017, pp. 593–597.

[11] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, Oct 2011, pp. 380–388.

[12] J. Erickson, *Hacking the art of exploitation*, 2nd ed. San Francisco, Calif: No Starch Press, 2008.

[13] T. Proffitt, *How Can You Build and Leverage SNORT IDS Metrics to Reduce Risk?* SANS Institute, Sep 2013. [Online]. Available: https://www.sans.org/reading-room/whitepapers/tools/paper/34350

[14] B. Vitalik, "A next-generation smart contract and decentralized application platform," Ethereum, 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[15] A. Morgan and T. Kukuk. (2010) The Linux-PAM system administrators' guide, version 1.1.2. [Online]. Available: http://www.linux-pam.org/Linux-PAM-html/Linux-PAM_SAG.html